Wang D, Pan JS, Tang JH. Single image deraining using residual channel attention networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 38(2): 439-454 Mar. 2023. DOI: 10.1007/s11390-022-0979-2

Single Image Deraining Using Residual Channel Attention Networks

Di Wang (王　迪), Jin-Shan Pan* (潘金山), *Member, IEEE*, and
Jin-Hui Tang (唐金辉), *Distinguished Member, CCF, Senior Member, IEEE, Member, ACM*

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

E-mail: wangdi_1211@njust.edu.cn; jspan@njust.edu.cn; jinhuitang@njust.edu.cn

Received September 9, 2020; accepted September 23, 2022.

Abstract Image deraining is a highly ill-posed problem. Although significant progress has been made due to the use of deep convolutional neural networks, this problem still remains challenging, especially for the details restoration and generalization to real rain images. In this paper, we propose a deep residual channel attention network (DeRCAN) for deraining. The channel attention mechanism is able to capture the inherent properties of the feature space and thus facilitates more accurate estimations of structures and details for image deraining. In addition, we further propose an unsupervised learning approach to better solve real rain images based on the proposed network. Extensive qualitative and quantitative evaluation results on both synthetic and real-world images demonstrate that the proposed DeRCAN performs favorably against state-of-the-art methods.

Keywords deraining, deep convolutional neural network (DCNN), channel attention, detail restoration, unsupervised finetuning

1 Introduction

Images taken in a rain environment are usually degraded by rain. Such images usually affect the high-level vision tasks, e.g., object detection[1], image recognition[2], and semantic segmentation[3, 4]. Thus, it is a great of interest to remove rain from the rain images. The goal of image deraining is to recover a clear image from a rain image. It is a highly ill-posed problem as the degradation process is complex and only the rain image is known. To solve this problem, many kinds of image priors have been proposed, such as sparse prior[5–9], layer prior[10], and Gaussian Mixture Model (GMM) prior[11], to make the problem well-posed. These methods are effective for images with little rain. However, they usually involve complex designs of image priors and are less effective when images contain heavy rain.

To avoid developing hand-crafted priors and better capture the inherent properties of clear images, numerous deep learning based methods have been proposed[12–16]. These deep learning based methods usually perform better than conventional hand-crafted prior based methods by large margins. However, most of them have poor estimations of structural details. To overcome this problem, several methods[14, 15, 17] introduce an image decomposition model into deep convolutional neural networks (DCNNs) to estimate structural details and remove rain streaks. However, it is impractical to decompose the background and rain layers accurately. Generally, the decomposed rain layer either contains some background details or underestimates the rain.

To further improve the performance of image deraining, some methods focus on feature representation. Yang *et al.*[18] proposed an effective method to combine rain detection and rain removal in a unified DCNN. Fu *et al.*[19] proposed a deep detail network, which introduces negative residual mapping to obtain high-frequency detail features. Li *et al.*[16] and Fu *et al.*[20] proposed a non-locally enhanced encoder-decoder network and a Gaussian-Laplacian pyramid

Regular Paper

This work was supported by the National Key Research and Development Program of China under Grant No. 2018AAA0102001 and the Fundamental Research Funds for the Central Universities of China under Grant No. 30920041109.

*Corresponding Author

©Institute of Computing Technology, Chinese Academy of Sciences 2023

model respectively to model multi-scale feature representation. Chen *et al.*[21] proposed a gated context aggregation model, which uses dilated convolutions to expand the feature receptive field. Fu *et al.*[22] proposed a deep tree structured fusion network to aggregate features to reconstruct rain-free images. Ren *et al.*[23] and Li *et al.*[24] proposed recursive neural networks to refine learned features, and then remove rain streaks completely. As demonstrated in [25], these methods may struggle to handle kinds of rain streaks. Subsequently, Zhang and Patel[25] developed an effective density-aware image deraining method and achieved decent results. However, it does not distinguish image structures and details, which has a poor effect on detail estimation. These existing image deraining methods have a poor effect on detail restoration.

To restore a clear image with fine details, motivated by that the attention mechanism can model the most important structures and details of an image, we develop an effective method based on a deep neural network with a residual channel attention mechanism, namely DeRCAN. The attention mechanism is applied to the feature space of images. By doing so, the proposed method is able to facilitate more accurate estimations of structures and details in the image deraining. In addition, to generate realistic images, we develop a perceptual constraint in the feature space. Although significant improvement has been made, the above methods tend to overfit the training data and fail to generalize well in real-world applications. To better solve real-world rain images, we develop an unsupervised finetuning method based on the proposed DeRCAN. The main contributions of our work are summarized as follows.

- We propose an effective image deraining method based on a deep neural network with the residual channel attention mechanism. The attention mechanism is applied to the feature space to capture the most important features for image deraining.
- We analyze the effectiveness of the channel attention mechanism and develop a perceptual constraint in the feature space, and we use a residual learning method to model the structural details and generate realistic images.
- We propose an unsupervised finetuning method based on the proposed DeRCAN and exploit an unsupervised loss to better handle real-world rain images.
- We train the proposed DeRCAN in an end-to-end manner and discuss the effectiveness of key components in removing rain streaks. Extensive experimental results show that our method performs favorably against state-of-the-art methods on both benchmark datasets and real-world images.

The rest of this paper is organized as follows. Section 2 introduces related work including the deraining methods based on image priors and DCNNs. The supervised learning details of the proposed method on synthetic benchmarks are presented in Section 3. Section 4 presents the unsupervised finetuning of the proposed method in real-world rain scenes. In Section 5, we compare the proposed method with 11 state-of-the-art image deraining methods and implement sufficient ablation studies to verify the effectiveness of each component. Section 6 summarizes the main contents of the paper.

2 Related Work

Recent years have witnessed significant progress in image deraining due to the use of kinds of image priors[5, 6, 11] and deep neural networks[12–16]. In this section, we briefly review the most related work and put this work in proper context.

2.1 Rain Removal Based on Image Priors

Luo *et al.*[6] proposed a discriminative sparse coding for single image deraining. Zhu *et al.*[7] separated the rain layers from rain images based on the sparse representation, the rain streak directions, and the rain streak layer priors. Wang *et al.*[8] utilized a quasi-sparse prior to detect rain streaks and separate rain layers from backgrounds. Deng *et al.*[9] built a mathematical global sparse model for deraining. Du *et al.*[10] used the low correlation between background and rain streak images in the gradient domain to separate clean rain-free backgrounds from rain images. Li *et al.*[11] used layer priors to restore the background layer and the rain streak layer.

2.2 Rain Removal Based on DCNNs

Motivated by the success in high-level vision tasks, deep learning has been developed to solve the image deraining problem. Fu *et al.*[12] proposed a method called DerainNet, which is the first CNN-based deraining method. Pan *et al.*[13] proposed a dual convolution neural network (DualCNN) to jointly restore details and structures from rain images. Wang *et al.*[14] learned motion blur kernels to guide the synthesis of rain streak masks, and then used them to subtract from input images to obtain clean backgrounds. Ye *et al.*[15] proposed a deep symmetry enhanced network (DSEN) to model rain streaks in different directions so as to further help remove complex rain streaks in images. Li *et al.*[16] proposed a non-locally enhanced encoder-decoder network (NLEDN) which not only enables to model rain streaks accurately but also can preserve details. Li *et al.*[17] proposed a deep decomposition composition (DDC) network, which first decomposes the rain image into a background layer and a rain streak layer, and then reconstructs them, respectively. The final generated streak layer is fused with the background layer to form a new rain image, and the rain image is constrained by the input rain image to promote the generation of the intermediate deraining image. Yang *et al.*[18] extracted a binary mask to detect the position and shape of rain streaks, thus restoring a clean background image. Fu *et al.*[20] introduced a Gaussian-Laplacian pyramid model into the CNNs and proposed a lightweight pyramid network (LPNet) for single image deraining. Chen *et al.*[21] proposed a gated context aggregation network (GCANet), and Fu *et al.*[22] proposed a deep tree-structured fusion network. They used dilated convolutional neural networks for single image deraining. Li *et al.*[24] proposed a recursive context aggregation network called RESCAN, consisting of the squeeze-and-excitation blocks to remove rain streaks in a stage-by-stage manner. Besides, in addition to traditional CNNs, the generative adversarial networks (GANs) are also applied to the image deraining tasks (e.g., [26–29]). However, it is difficult to generate high-quality images using the relatively shallow CNNs, while the deeper neural networks cannot be easily solved. Fortunately, this problem has been further improved with the advent of the residual network (ResNet) by [30].

ResNet allows the deep neural networks to be trained easily. In some low-level vision tasks, ResNet is able to preserve more abundant details[31]. Motivated by the success of ResNet, significant progress has been made in image deraining. Fu *et al.*[19] proposed a deep detail network (DDN), which uses ResNet as its parameter layers and introduces negative residual mapping into the network to obtain high-frequency details. Ren *et al.*[23] considered network architecture, input and output, and loss functions, and then proposed a progressive recurrent network (PReNet) for single image deraining. Zhang and Patel[25] focused on the rain-density levels and proposed a density-aware multi-stream dense network (DID-MDN) for image deraining. The residual-aware classifier of DID-MDN generates rain-density labels to guide the multi-stream dense network to remove rain streaks. Fan *et al.*[32] proposed a residual guidance network (RGN), which uses the features from shallow residual blocks to guide deeper ones to obtain more accurate details.

In recent years, weakly-supervised image deraining methods have been proposed. Wei *et al.*[33] proposed a semi-supervised transfer learning framework to utilize simultaneously supervised and unsupervised knowledge for image deraining. Lin *et al.*[34] proposed a weakly-supervised deraining method based on knowledge distillation. These two methods require only unpaired rainy and clean images to generate supervision for the restoration of rain-free images. Although significant improvement has been made, the aforementioned networks tend to over-fit the training data. They are not generalized well in the real-world applications.

3 Proposed Method

3.1 Network Architecture

The flowchart of the proposed DeRCAN is shown in Fig.1. It contains three basic modules: the feature extraction module, feature refinement module, and image reconstruction module. The feature extraction module is composed of a single convolutional layer, which extracts features from the input image. The core of the proposed network is the feature refinement module. The feature refinement module contains a series of cascaded residual blocks, a feature fusion submodule, and a long skip connection. The residual block is composed of two convolution layers, a rectified linear unit (ReLU) layer, a skip connection, and an element-wise operation. Using a residual block can enlarge the receptive field of the network and avoid gradient vanishing. In addition, the skip connection in the residual block directly transfers features to the output, thus avoiding information loss. To better explore useful features for image deraining, we further develop the channel attention (CA) module described in Subsection 3.2 and embed it into each residual block called channel attention residual module (CARM), which enables useful features to be obtained. Note that we can use more than one feature refinement module in the proposed network (denoted

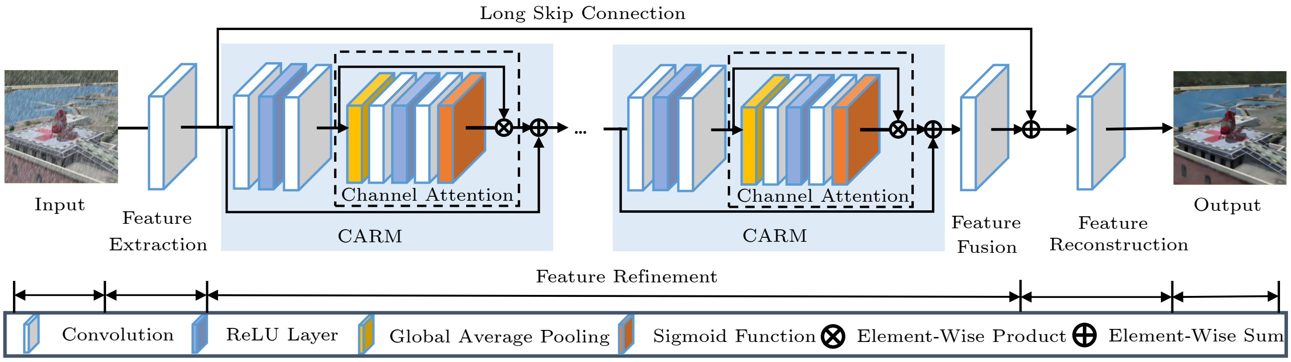


Fig.1. Architecture of the proposed DeRCAN. It consists of three modules: feature extraction, feature refinement, and image reconstruction. The network can be used to remove rain streaks on synthetic or real rain images. The network structure shows a supervised learning process on synthetic rain datasets.

as FRM). In Subsection 5.3, the effect of the number of the FRMs on performance is analyzed. After the feature refinement module, the feature fusion module is developed to fuse the features generated by CARMs. In addition, in order to prevent information loss during the feature refinement process, we use a long skip connection to transfer the features extracted by the feature extraction module to the image reconstruction module for better image restoration. The image reconstruction module is a single convolutional layer as well, and it converts the refined features into an RGB rain-free image reversely. The proposed network is trained in an end-to-end fashion and can be achieved in a supervised learning manner (Subsection 3.3) or an unsupervised finetuning manner (Section 4).

3.2 CA Module

Motivated by the SENet[35] and CBAM[36], we intend to use the channel attention mechanism to refine the features by exploiting the channel-wise interdependencies. As shown in Fig.1, the CA module consists of a global average pooling layer, two convolution layers, a ReLU layer, and a sigmoid function layer. The global average pooling layer aims to squeeze the feature maps with spatial dimension $H \times W \times C$ into a feature label sequence with dimension $1 \times 1 \times C$; therefore it represents the global receptive field. Here H and W are the height and the width of the feature map respectively, and C refers to the number of the feature channels. Suppose we input C feature maps $F = (f_1, ..., f_m, ..., f_C)$ with the size of $H \times W$, then the m-th label l_m corresponding to f_m can be obtained by (1).

$$l_m = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_m(i, j), \tag{1}$$

where $f_m(i, j)$ denotes the m-th feature map at position (i, j). We obtain a channel-wise feature sequence $L = (l_1, l_2, ..., l_m, ..., l_C), L \subseteq \mathbb{R}^C$. Then, we use the downscaling convolutional layer with a weight set w_{Down} and a reduction factor $reduction = 16$ to downscale the global sequence L nonlinearly (shown in (2)).

$$\hat{L}_{\text{Down}} = ReLU(Conv_{\downarrow}(L, w_{\text{Down}}, 1/reduction)), \tag{2}$$

where $Conv_{\downarrow}$ denotes the downscaling convolutional layer with a filter size of 1×1. $ReLU$ denotes the ReLU activation function.

We then use the upscaling convolutional layer with a weight set w_{Up} and $reduction = 16$ to upscale the sequence $\hat{L}_{\text{Up}}$. Continuously, we exploit a simple gating mechanism with a sigmoid function to model the channel-wise interdependencies and capture a new channel-wise feature sequence $\hat{L}$ (shown in (3)).

$$\begin{aligned} \hat{L}_{\text{Up}} &= Conv_{\uparrow}(\hat{L}_{\text{Down}}, w_{\text{Up}}, reduction), \\ \hat{L} &= Sigmoid(\hat{L}_{\text{Up}}), \end{aligned} \tag{3}$$

where $Conv_{\uparrow}$ denotes the upscaling convolutional layer with a filter size of 1×1 and $Sigmoid$ denotes the sigmoid gating function. Finally, we can update the features as shown in (4).

$$\hat{f}_m = \hat{l}_m \otimes f_m, \tag{4}$$

where $\otimes$ denotes the channel-wise multiplication between the feature map $f_m \subseteq \mathbb{R}^{H \times W}$ and the channel-wise feature $\hat{l}_m \subseteq \mathbb{R}^{1 \times 1}$.

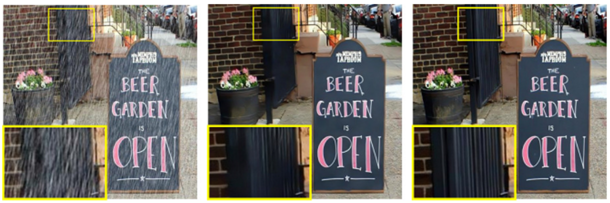
In this way, the network enables to increase the sensitivity to useful information and suppress useless information adaptively by utilizing the channel-wise interdependencies, thus making the network focus on more valuable features. We will demonstrate the effectiveness of the CA mechanism in Subsection 5.3.

3.3 Supervised Learning

Let $\{\hat{x}_i\}_{i=1}^{N_s}$ and $\{x_i\}_{i=1}^{N_s}$ denote the predicted images by the supervised learning approach and the corresponding clear images, respectively. We use the pixel-wise content loss function (shown in (5)) to constrain the network.

$$\mathcal{L}_1 = \sum_{i=1}^{N_s} \|\hat{x}_i - x_i\|_1 \,, \tag{5}$$

where N_s denotes the number of synthetic training images and the L_1 norm is used. We note that (5) is able to remove artifacts but tends to over-smooth image details (see Fig.2(b)).



(a) (b) (c)

Fig.2. Effect of the loss functions on image deraining. (a) Input. (b) Results without $\mathcal{L}_\text{p}$. (c) Results with $\mathcal{L}_\text{p}$.

To generate more realistic images, we further develop a perceptual loss (shown in (6)) based on the features that are estimated by the VGG[37] network pre-trained on ImageNet[38].

$$\mathcal{L}_\text{p} = \sum_{i=1}^{N_s} \left\|\psi_j^k(\hat{x}_i) - \psi_j^k(x_i)\right\|_2^2 , \tag{6}$$

where $\psi_j^k(\hat{x})$ and $\psi_j^k(x)$ denote the k-th feature maps of $\hat{x}$ and x extracted from the j-th layer of the VGG-19 network, respectively. Based on above considerations, the loss function used for the supervised training is defined as (7).

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_\text{p}, \tag{7}$$

where λ is a positive weight parameter. The effect of the perceptual loss is analyzed in Subsection 5.3.

4 Unsupervised Finetuning

We note that existing image deraining methods based on deep neural networks tend to overfit the training datasets and cannot generalize well on real rain images. To overcome this problem, we propose an unsupervised finetuning method based on the proposed DeRCAN. As the ground truths of real rain images are not available, using the loss functions proposed in Subsection 3.3 to constrain the proposed network is not feasible. We note that the total variational (TV)[39] regularization as an effective image prior is able to model the distribution of clear image gradients. We use it to constrain the network to handle real-world images. The TV regularization is defined as:

$$\mathcal{L}_\text{TV} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left(\|\partial_h \hat{x}_i\|_2 + \|\partial_v \hat{x}_i\|_2\right). \tag{8}$$

Here $\{\hat{x}_i\}_{i=1}^{N_r}$ denotes the network outputs. N_r denotes the number of real training images. ∂_h and ∂_v denote horizontal and vertical gradient operators, respectively. Note that we only use (8) to constrain this unsupervised learning method. The effectiveness of the proposed unsupervised finetuning method is discussed in Subsection 5.3.

5 Experimental Results

In this section, we first present details about the training and testing datasets and the parameter settings. Then, we analyze the effect of each component of the proposed DeRCAN on image deraining. Finally, we evaluate the proposed method against state-of-the-art methods including: DerainNet[12] (TIP 2017), JORDER[18] (TPAMI 2020), DDN[19] (CVPR 2017), RGN[32] (ACM MM 2018), LPNet[20] (TNNLS 2019), NLEDN[16] (ACM MM 2018), RESCAN[24] (ECCV 2018), DID-MDN[25] (CVPR 2018), GCANet[21] (WACV 2019), PReNet[23] (CVPR 2019), SPANet[40] (CVPR 2019), and Rain O'er Me[34] (TIP 2020), and show the qualitative and quantitative results. We use the peak signal-to-noise ratio (PSNR)[41] and structural similarity (SSIM)[42] as metrics.

5.1 Datasets

Synthetic Datasets. Table 1 shows a description of the datasets used for training and testing. The DID-

Table 1. Description of the Datasets

Dataset	Number of Training Images	Number of Testing Images	Label
Rain100L[18]	1 800	200	Yes
Rain100H[18]	1 800	200	Yes
Rain14000[19]	12 600	1 400	Yes
DID-MDN[25]	12 000	1 200	Yes
Real-world	300	50	No

MDN training dataset contains 12 000 images degraded by medium rain, heavy rain, and light rain, respectively. During the training phase, we randomly select 11 400 images from the DID-MDN training dataset for training and use the remaining 600 images for validation. During the testing phase, we use 1 200 images from the DID-MDN testing dataset to measure the performance of the proposed DeRACN.

Real-World Datasets. We collect 300 real-world rain images from the Internet for the training process of unsupervised learning. We use 50 images for testing.

5.2 Network Parameters and Training Settings

Network Parameters. Our network is trained in two manners: supervised learning and unsupervised finetuning, which share the same network parameters. The filter size of all convolutional layers is 3 × 3, except for the upscaling and downscaling convolutional layers in the CA module of the network. For the feature refinement module, we use 20 residual blocks embedded in the CARM in each FRM. The reduction factor is 16.

Training Settings. During the training, the batch size is 16 and the patch size is 32 × 32. The number of feature channels is 256. The learning rate is initialized to 10^{-4} and decreases to half every 10 epochs during the training. The optimizer is adaptive moment estimation (Adam)[43], and its parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. We train 200 epochs with momentum of 0.9. For the supervised learning, we use (7) to constrain the network. According to the analysis in Table 2, we empirically set λ to 5.0. For the unsupervised finetuning, we finetune the proposed network trained on the DID-MDN[25] dataset with the constraint (8) using the real-world training set described in Table 1. The other settings are the same as those of the supervised learning.

Table 2. Analysis for Weight Parameter λ of Perceptual Loss Function $\mathcal{L}_{\mathrm{p}}$

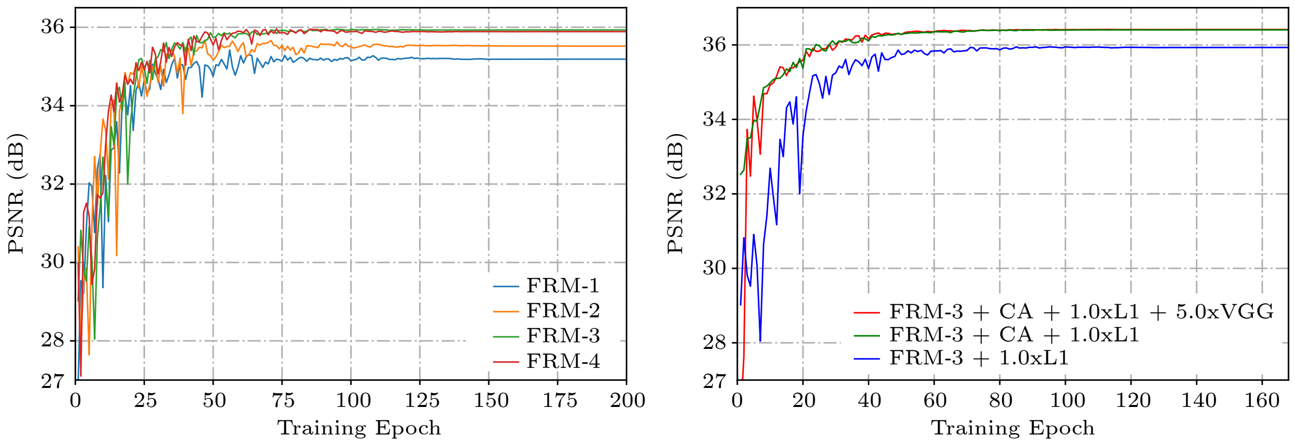
λ	PSNR	SSIM
1.0	34.18	0.929 5
5.0	**34.24**	**0.930 2**
10.0	34.20	0.930 2
50.0	34.07	0.929 9
100.0	34.04	0.929 4

Note: The best results are displayed in bold.

5.3 Ablation Studies

We examine the effectiveness of each component of our DeRCAN and train the baselines on the DID-MDN dataset[25] with the same settings for fair comparisons. Fig.3 and Table 3 provide comprehensive experiment results in training and evaluation aspects, respectively. In Table 3, B_a, B_b, ..., B_f refer to the experiments conducted for ablation analysis.

Effectiveness of the Number of FRMs. To extract more useful features for image deraining, we use several FRMs in the proposed network. Fig.3(a) shows that a higher PSNR tendency is obtained using more FRMs in our network during the training process. However, the result of using four FRMs is basically the same as that of using three FRMs in the network. Consequently, considering the size and parameters of



(a) (b)

Fig.3. Training convergence on PSNR of our method. (a) Comparisons of our method with different numbers of FRMs. (b) Influence of CA mechanism and loss functions. "FRM-N" denotes that we use N FRMs in the proposed DeRCAN.

Table 3. Ablation Analysis of the Proposed Method on the DID-MDN Dataset

Method	FRM-1	FRM-3	CA	$\mathcal{L}_{\rm p}$	PSNR	SSIM
B_a	✓				32.14	0.917 3
B_b	✓		✓		33.78	0.925 5
B_c	✓		✓	✓	33.93	0.926 9
B_d		✓			32.19	0.920 6
B_e		✓	✓		34.11	0.929 6
B_f		✓	✓	✓	**34.24**	**0.930 6**

Note: The best results are displayed in bold.

the network model, we finally decide to use three FRMs to perform the image deraining task. Three groups of contrast experiments (B_a, B_d), (B_b, B_e) and (B_c, B_f) in Table 3 show that PSNR values of using three FRMs are 0.05 dB to 0.33 dB higher than those of using one FRM.

Loss Functions. We use L_1 norm as the fundamental pixel-level content loss function of the proposed network. To constrain the network in the feature space so as to learn better parameters, we introduce the perceptual loss function $\mathcal{L}_{\rm p}$ described in (6). Fig.3(b) also shows that the PSNR tendency of the network with $\mathcal{L}_{\rm p}$ is slightly higher than that of the network without $\mathcal{L}_{\rm p}$ during the training. Correspondingly, two groups of contrast experiments (B_b, B_c) and (B_e, B_f) in Table 3 show that using (6) will increase the PSNR by 0.13 dB to 0.15 dB. Visually, Fig.2(c) demonstrates that using perceptual loss is able to recover more structural details.

Adversarial Loss. To explain the effect of GANs for image deraining clearly, we introduce adversarial loss ($\mathcal{L}_{\rm adv}$) to constrain our network. Table 4 shows the quantitative results of the models constrained by different loss functions. We find the results with $\mathcal{L}_{\rm adv}$ have lower PSNR and SSIM values than our method. This is because $\mathcal{L}_{\rm adv}$ constrains the network to achieve finer and more realistic details that approximate the true data distribution. Although the visual effect looks more realistic, noise and artifacts in the smooth regions of the images are also introduced. As shown in Fig.4, texture regions in Fig.4(b) and Fig.4(c) contain more structural details than those in Fig.4(c), while there exist more noise and artifacts in the smooth regions.

Table 4. Quantitative Results of the Models Constrained by Different Loss Functions on DID-MDN Dataset

	PSNR	SSIM
$\mathcal{L}_1$	32.19	0.920 6
$\mathcal{L}_1$+$\mathcal{L}_{\rm adv}$	33.81	0.929 5
$\mathcal{L}_1$+$\mathcal{L}_{\rm adv}$+$\mathcal{L}_{\rm p}$	34.12	0.929 8
$\mathcal{L}_1$+$\mathcal{L}_{\rm p}$ (ours)	**34.24**	**0.930 2**

Note: The best results are displayed in bold.

CA Module. We conduct ablation studies about the CA module to demonstrate its effectiveness. Two groups of contrast experiments (B_a, B_b) and (B_d, B_e) in Table 3 show that using the CA module increases the PSNR value by 1.6 dB to 2.0 dB and the SSIM values by 0.01 approximately. During the training, the tendencies of B_d and B_e are shown with blue and green lines in Fig.3(b). Obviously, the network with the CA module obtains higher quantitative results. In addition, we provide feature visualizations in Fig.5,

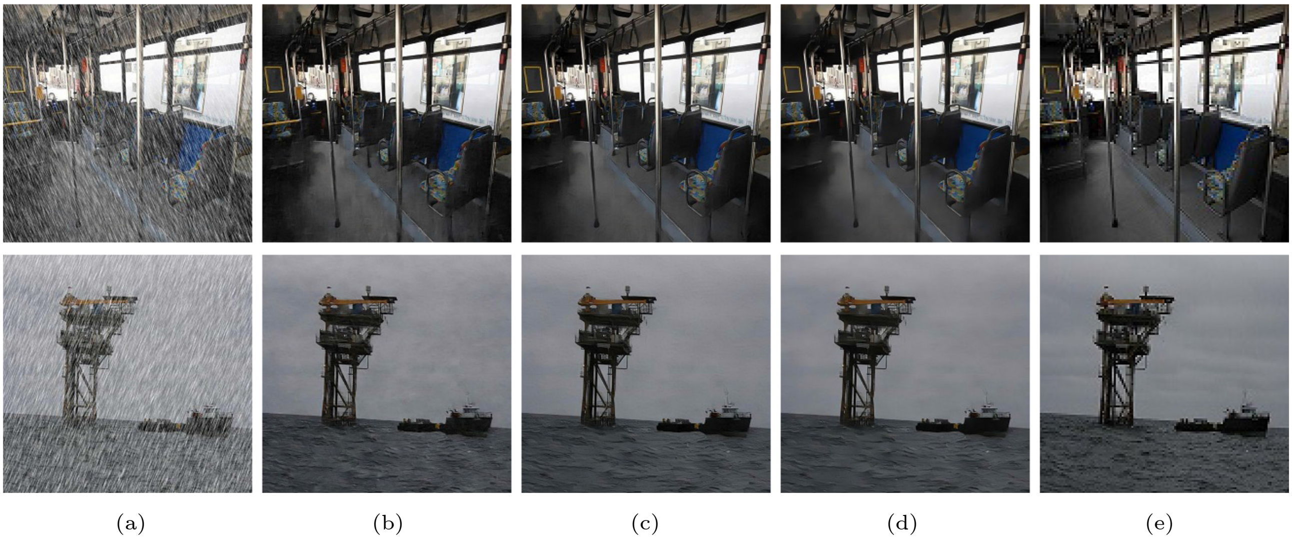


Fig.4. Visualizations of removing rain streaks by the models constrained by different loss functions on the heavy rain images. (a) Input. (b) $\mathcal{L}_1$+$\mathcal{L}_{\rm adv}$. (c) $\mathcal{L}_1$+$\mathcal{L}_{\rm adv}$+$\mathcal{L}_{\rm p}$. (d) $\mathcal{L}_1$+$\mathcal{L}_{\rm p}$. (e) Ground truth.

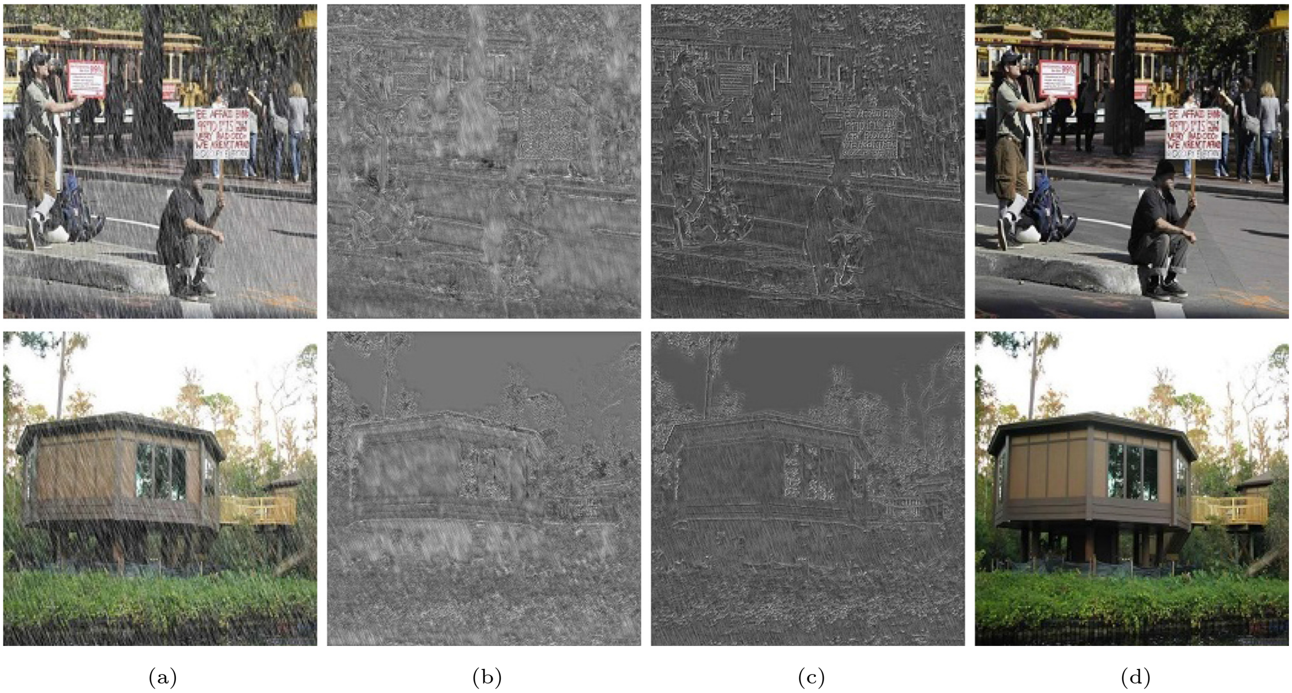


Fig.5. Effectiveness of the CA module. (a) Input. (b) Feature maps generated by the model without the CA module. (c) Feature maps generated by the model with the CA module. (d) Ground truth.

which are generated by the models with or without the CA module. We train the network with only one FRM and visualize feature maps generated by the last CARM or residual block at the same epoch during training. Fig.5(b) shows blurry edges and textures and contains a great quantity of useless information. In contrast, Fig.5(c) shows clear edges and textures, which are beneficial to reconstruct high-quality images. These comparisons prove that using the CA module contributes to an improvement of the proposed method.

Downscaling and Upscaling in the CA Module. The utilities of the downscaling and upscaling operations are motivated by SENet[35]. These two operations are used to generate channel-wise scaling factors of the global sequence to extract more useful features that contribute to clear image restoration. To evaluate the effect of the downscaling and upscaling operations, we disable them in the proposed method and train this baseline method using the same settings. The quantitative comparison results on the DID-MDN dataset in Table 5 show that using the downscaling and upscaling operations is able to improve the performance of the image deraining. We further visualize the feature maps of residuals learned by the proposed method without the downscaling and upscaling operations. In Fig.6, the model without downscaling and upscaling pays the same attention to

Table 5. Effectiveness of Downscaling (Down) and Upscaling (Up) Operations in the Global Sequence

	PSNR	SSIM
Without Down & Up	33.22	0.9251
With Down & Up (ours)	**34.24**	**0.9302**

Note: The best results are displayed in bold.

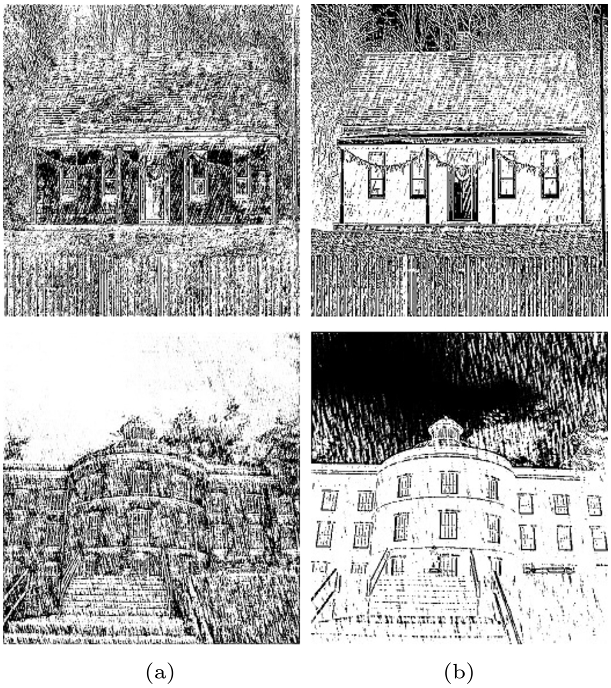
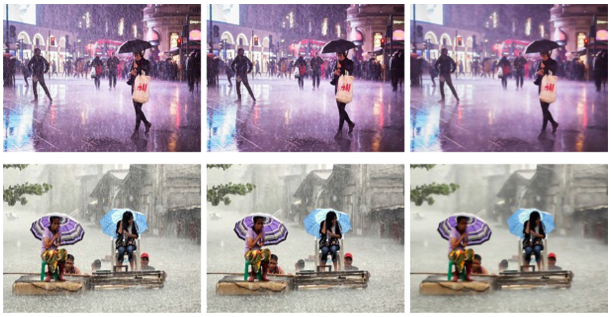


Fig.6. Feature visualizations of different models on heavy rain images. (a) Without Down & Up. (b) With Down & Up.

the foreground and background content, while the model with these two operations adaptively recalibrates channel-wise feature response and pays more attention to the details, which is beneficial to restoring clean images with more details.

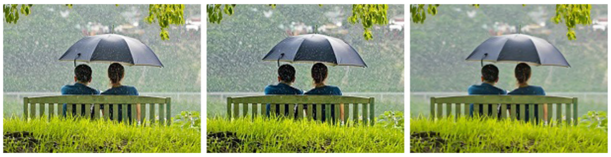
Unsupervised Finetuning. The aim of the unsupervised finetuning is to improve the performance of our DeRCAN in handling real-world rain images. Fig.7(b) shows that the proposed method trained on synthetic datasets through supervised learning cannot effectively remove rain streaks when handling real rain images. In contrast, our method learned on real rain images by unsupervised finetuning can remove rain streaks and generate better results as shown in Fig.7(c). Note that we adopt the model pretrained on the DID-MDN[25] dataset to implement unsupervised finetuning.



(a) (b) (c)

Fig.7. Effectiveness of the unsupervised learning on real-world image deraining. (a) Input. (b) Supervised model. (c) Unsupervised model.

TV Loss. Using TV loss is effective for unsupervised image deraining, as shown in Fig.7. We also examine another commonly-used loss function, i.e., dark channel (DC) loss[44]. In our experiments, we use the DC loss $\mathcal{L}_{\mathrm{DC}}$ instead of the TV loss $\mathcal{L}_{\mathrm{TV}}$ to implement the unsupervised deraining task on the real-world rain images. Fig.8 shows that using $\mathcal{L}_{\mathrm{DC}}$ can enhance the color contrast, the deraining images still contain rain streaks. In contrast, the proposed method with TV loss is able to remove rain streaks and generate better images.



(a) (b) (c)

Fig.8. Influence of different unsupervised loss functions. (a) Input. (b) Results under $\mathcal{L}_{\mathrm{DC}}$. (c) Results under $\mathcal{L}_{\mathrm{TV}}$.

5.4 Evaluations on Synthetic Datasets

In this subsection, we evaluate the proposed method quantitatively and qualitatively. To be fair, we retrain the deraining models of state-of-the-art methods for each dataset. We evaluate the deraining results on synthetic datasets with PSNR[41] and SSIM[42] metrics.

Firstly, we prove the effectiveness and advancement of the proposed method quantitatively. Table 6 shows the quantitative results compared with the

Table 6. Quantitative PSNR and SSIM Values of the Proposed Method and State-of-the-Art Methods on Four Different Datasets

Method	Rain100L		Rain100H		Rain14000		DID-MDN	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Input	26.71	0.844	13.08	0.373	25.23	0.790	23.63	0.732
DerainNet[12]	29.35	0.932	22.26	0.714	25.52	0.872	23.92	0.849
JORDER-E[18]	37.10	0.980	24.54	0.802	27.08	0.872	32.11	0.912
DDN[19]	34.41	0.958	26.13	0.803	27.61	0.901	30.99	0.886
RGN[32]	33.16	0.963	25.25	0.841	29.51	0.901	30.18	0.901
LPNet[20]	29.11	0.880	14.26	0.423	25.64	0.836	22.75	0.835
NLEDN[16]	36.57	0.975	**30.38**	0.894	29.79	0.898	33.16	0.919
RESCAN[24]	34.02	0.975	26.45	0.846	28.57	0.891	29.95	0.884
DID-MDN[25]	30.48	0.932	26.35	0.829	27.99	0.869	27.95	0.909
PReNet[23]	37.48	0.979	29.46	0.899	32.56	0.933	33.93	0.933
SPANet[40]	34.27	0.964	25.64	0.843	29.99	0.901	30.05	**0.934**
Rain O'er Me[34]	33.09	0.955	–	–	28.50	0.890	28.72	0.873
Our method	**39.63**	**0.986**	29.87	**0.906**	**33.07**	**0.933**	**34.24**	0.931

Note: The best results are displayed in bold.

state-of-the-art methods on datasets including Rain100L[18], Rain100H[18], Rain14000[19], and DID-MDN[25], respectively. The best results are highlighted. Apparently, our method achieves favorable performance on all four datasets. Then, we demonstrate the effectiveness of our method qualitatively. Due to the layout limitations, we select seven relatively favorable methods in Table 6 to compare visual results with the proposed method on the above four datasets. Fig.9, Fig.10, and Fig.11 show the deraining results on images of medium rain, heavy rain, and light rain from the DID-MDN testing dataset respectively, comparing the proposed method with DerainNet[12], DDN[19], RGN[32], DID-MDN[25], and GCANet[21] methods. Fig.12 shows the visual comparisons of the proposed method with DDN[19], SPANet[40], and PReNet[23] on the Rain100L and Rain100H datasets. Fig.13 shows the visual comparisons of the proposed method with DDN[19], SPANet[40], and PReNet[23] on the Rain14000 dataset. As can be seen, the deraining results by DerainNet, DDN, and RGN show obvious blurry marks, and the structures of images are dam-

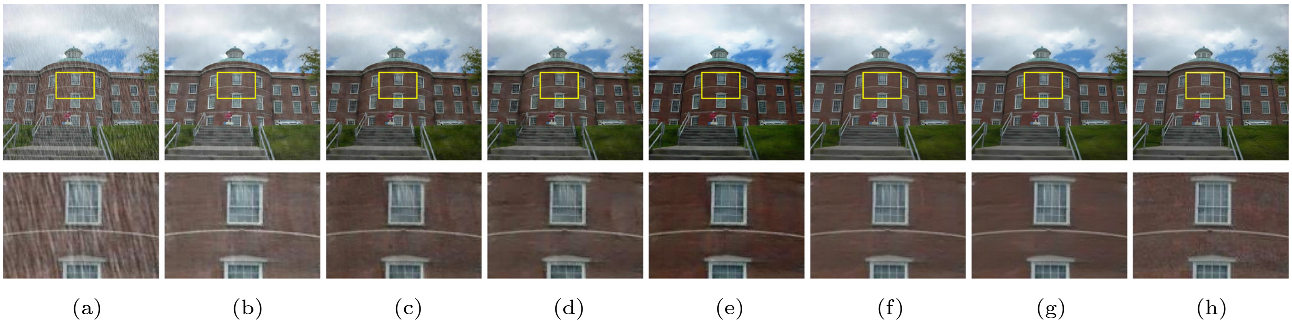


Fig.9. Deraining results on the DID-MDN dataset with medium rain density level. (a) Input (PSNR/SSIM: 22.17/0.730). (b) DerainNet[12] (PSNR/SSIM: 23.16/0.890). (c) DDN[19] (PSNR/SSIM: 31.96/0.910). (d) RGN[32] (PSNR/SSIM: 30.46/0.911). (e) DID-MDN[25] (PSNR/SSIM: 25.41/0.909). (f) GCANet[21] (PSNR/SSIM: 34.55/0.938). (g) Ours (PSNR/SSIM: 34.55/0.942). (h) Ground truth.

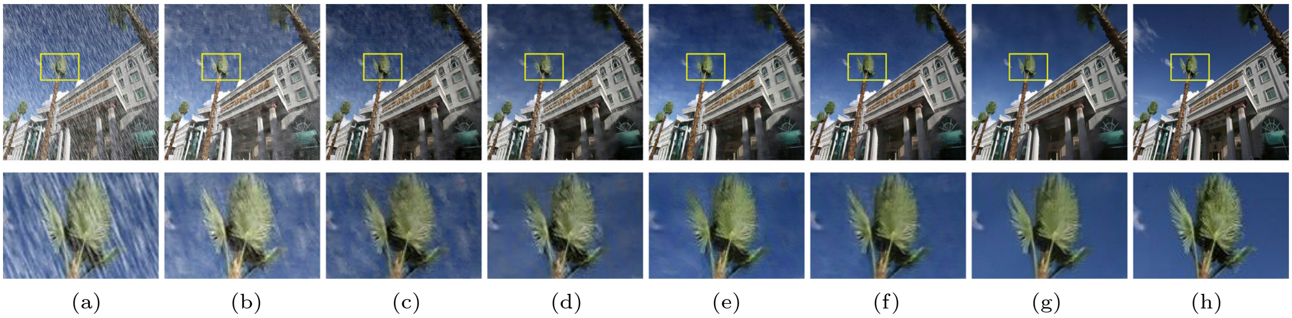


Fig.10. Deraining results on the DID-MDN dataset with heavy rain density level. (a) Input (PSNR/SSIM: 15.79/0.436). (b) DerainNet[12] (PSNR/SSIM: 17.17/0.704). (c) DDN[19] (PSNR/SSIM: 26.97/0.781). (d) RGN[32] (PSNR/SSIM: 27.05/0.861). (e) DID-MDN[25] (PSNR/SSIM: 27.18/0.858). (f) GCANet[21] (PSNR/SSIM: 29.56/0.884). (g) Ours (PSNR/SSIM: 29.93/0.902). (h) Ground truth.

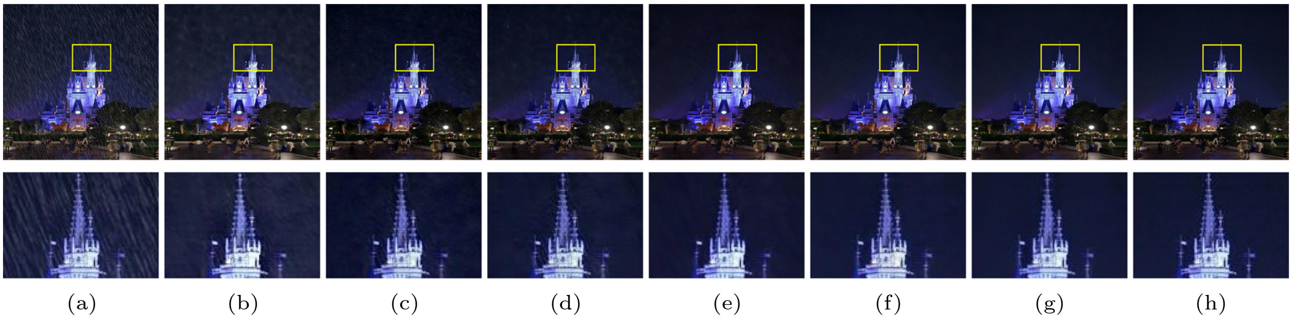


Fig.11. Deraining results on the DID-MDN dataset with light rain density level. (a) Input (PSNR/SSIM: 26.12/0.670). (b) DerainNet[12] (PSNR/SSIM: 27.57/0.896). (c) DDN[19] (PSNR/SSIM: 32.93/0.916). (d) RGN[32] (PSNR/SSIM: 34.91/0.944). (e) DID-MDN[25] (PSNR/SSIM: 34.89/0.950). (f) GCANet[21] (PSNR/SSIM: 36.73/0.958). (g) Ours (PSNR/SSIM: 37.58/0.962). (h) Ground truth.

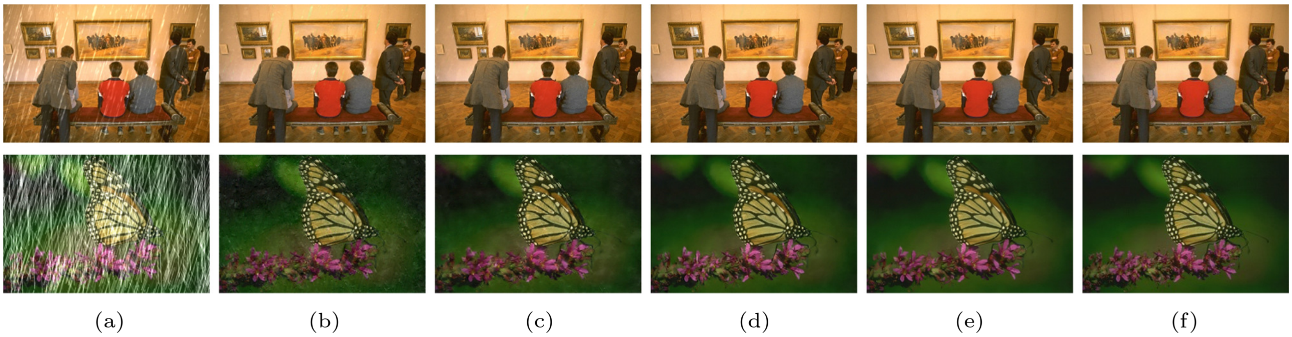


Fig.12. Deraining results on the Rain100L and Rain100H datasets. (a) Input (PSNR/SSIM: 27.85/0.836 & 12.32/0.199). (b) DDN[19] (PSNR/SSIM: 37.02/0.962 & 29.22/0.801). (c) SPANet[40] (PSNR/SSIM: 37.10/0.970 & 30.63/0.907). (d) PReNet[23] (PSNR/SSIM: 40.34/0.981 & 33.24/0.949). (e) Ours (PSNR/SSIM: 42.49/0.988 & 34.95/0.956). (f) Ground truth.

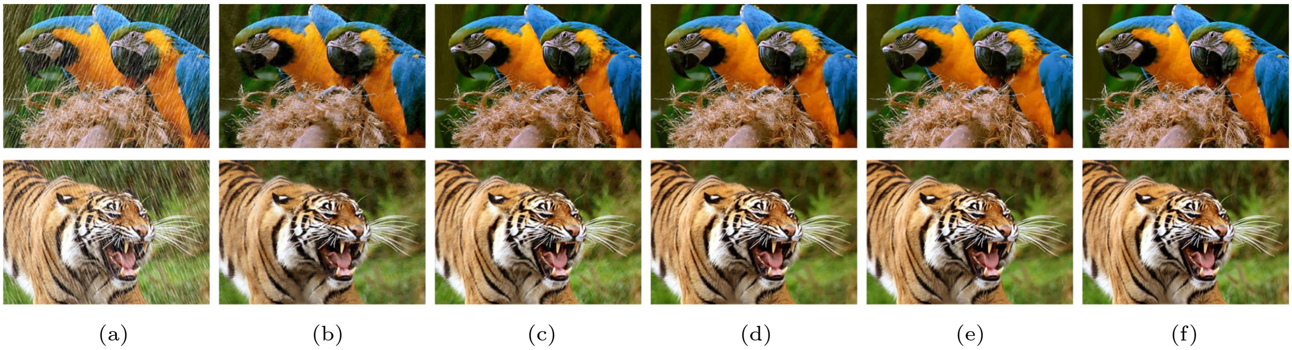


Fig.13. Deraining results on the Rain14000 dataset. (a) Input (PSNR/SSIM: 19.67/0.622 & 20.71/0.688). (b) DDN[19] (PSNR/SSIM: 28.34/0.847 & 28.65/0.872). (c) SPANet[40] (PSNR/SSIM: 30.69/0.929 & 29.70/0.904). (d) PReNet[23] (PSNR/SSIM: 31.21/0.932 & 33.24/0.949). (e) Ours (PSNR/SSIM: 31.41/0.931 & 34.95/0.956). (f) Ground truth.

aged. The results of DID-MDN show color distortion and some artifacts. Our results are most similar to the ground truth. Therefore, it is proved that the proposed method performs favorably against state-of-the-art methods on synthetic datasets.

5.5 Evaluations on Real-World Dataset

We use some real-world rain images to evaluate the effectiveness of the proposed method. We compare the proposed method with some state-of-the-art methods including NLEDN[16], DID-MDN[25], GCANet[21], and PReNet[23]. Fig.14 shows visual comparisons on real-world rain images. Our method performs more favorably than the state-of-the-art methods. These results indicate that the proposed method is favorable to deal with rain images in the real world.

5.6 Removing Haze-Like Effect on Heavy Rain Images

The proposed method directly restores clean images from given rain images in an end-to-end manner and is able to deal with heavy rain images with a haze-like effect. To evaluate our method in such a case, we use the Outdoor-Rain dataset[27] with a haze-like effect created by the data synthesis method in HRIR[45] to directly train the proposed deraining network. This dataset contains 9 000 rainy and clean image pairs for training and 1 500 rainy and clean image pairs for testing. Table 7 shows quantitative results of our method and HRIR on the heavy rain images. The proposed method performs better than HRIR by a large margin. In addition, Fig.13 shows that the deraining results generated by [37] still have artifacts and the haze-like effect is not completely eliminated, while the results of our method are much clearer. Fig.15 shows that our method performs better than HRIR on the heavy rainy images with a haze-like effect.

5.7 Applications on Video Deraining Task

Our method can be directly applied to videos by processing image deraining frame by frame. Although separately handling each frame alone[46] will destroy the temporal information of the videos, our experimental results in Fig.16 show that the proposed method is able to handle video deraining tasks and generate clear videos.

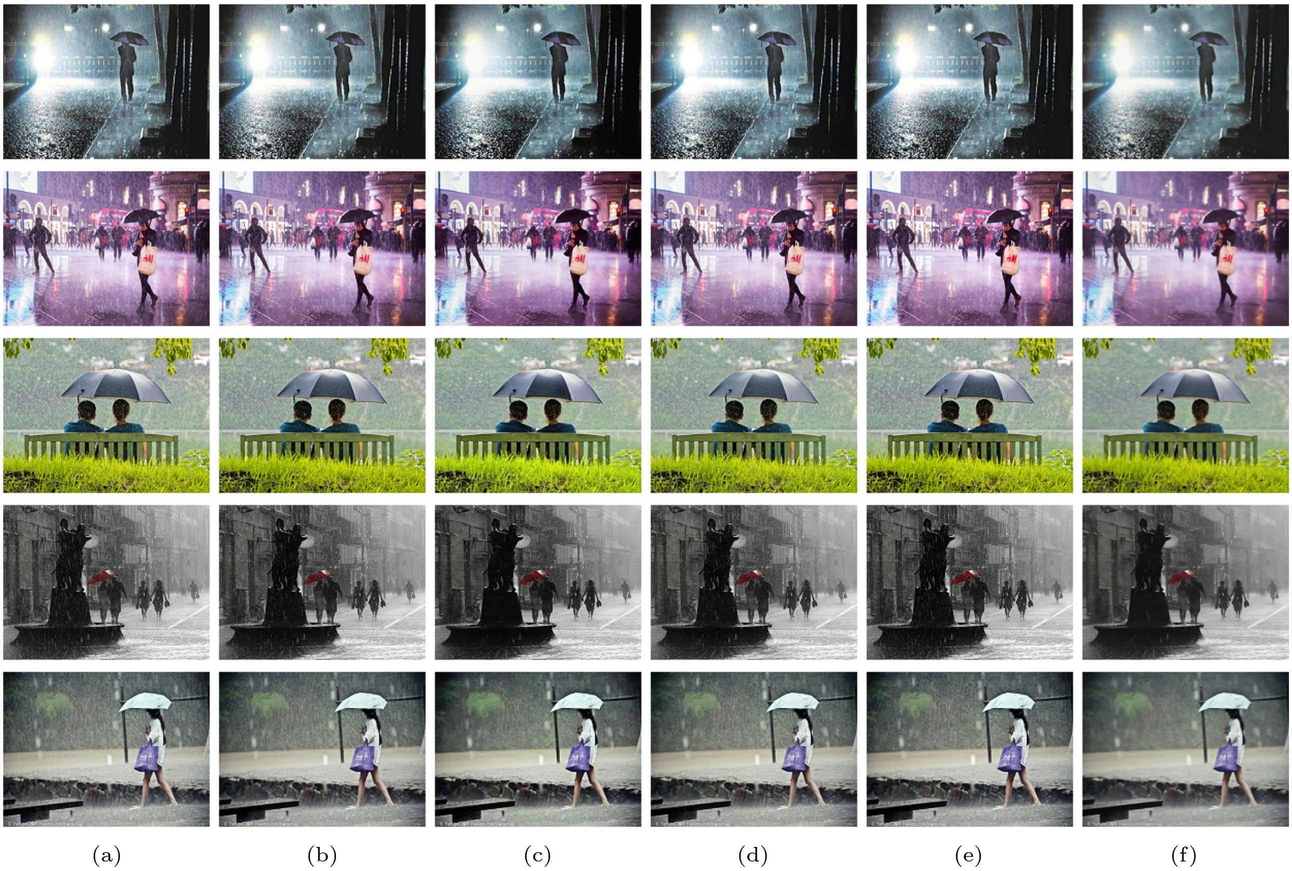


Fig.14. Visual comparisons of real-world rain images for single image deraining with several state-of-the-art methods. (a) Input. (b) NLEDN[16]. (c) DID-MDN[25]. (d) GCANet[21]. (e) PReNet[23]. (f) Ours.

Table 7. Quantitative Results of the Proposed Method and HRIR on the Heavy Rain Images

	PSNR	SSIM
Input	13.04	0.540
HRIR[45]	21.78	0.817
Ours	24.90	0.887

5.8 Applications on Object Detection Task

Since the rain streaks occlude and blur the content of images, which severely reduces the accuracy of object detection[47], it is unavoidable to use image deraining in the preprocessing process. We use a pre-

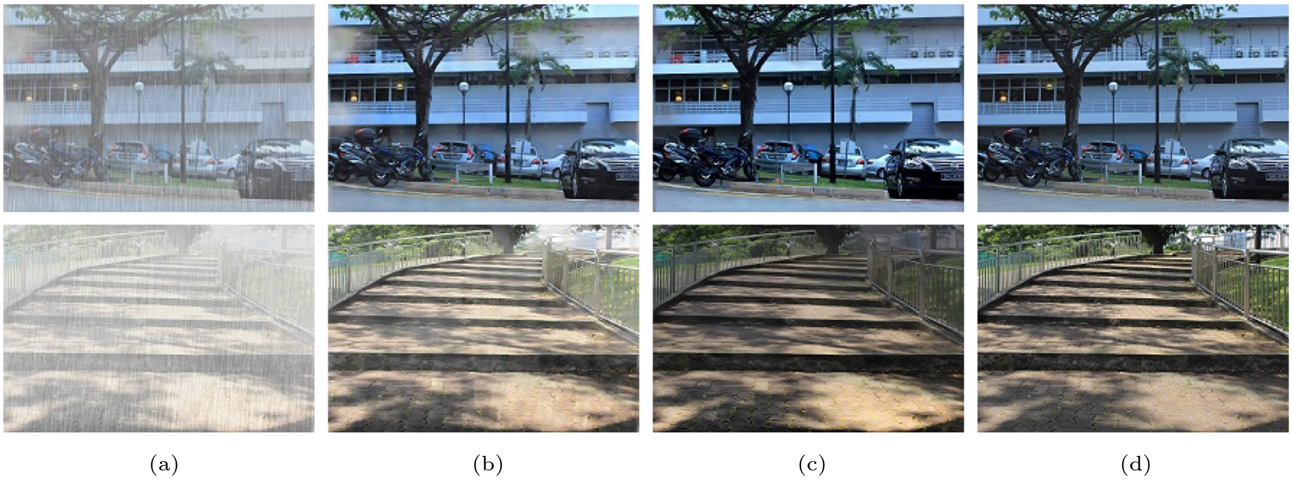


Fig.15. Visualizations of removing haze-like effect by the proposed method on the heavy rain images. (a) Heavy rain images with haze-like effect (PSNR/SSIM: 12.56/0.530 & 9.46/0.525). (b) Results of [45] (PSNR/SSIM: 21.22/0.805 & 16.51/0.743). (c) Results of our DeRCAN (PSNR/SSIM: 26.73/0.892 & 20.14/0.746). (d) Ground truth.

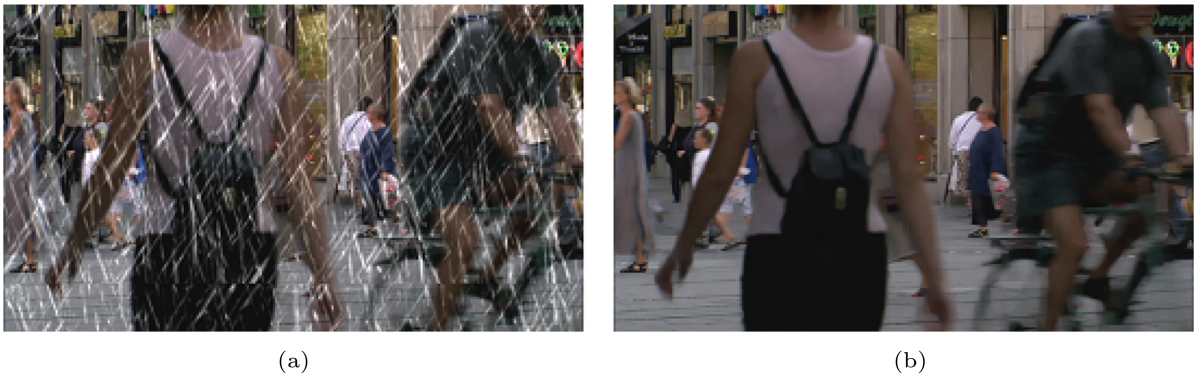


Fig.16. Deraining results predicted by the proposed method on the RainSynComplex25[40] video. (a) Rain video. (b) Ours. The corresponding complete videos are displayed online①.

trained Faster-RCNN[48] framework trained on the COCO[49] and VOC[50] datasets to detect objects in real rain images and deraining images, respectively. Fig.17 shows an object detection example. We can observe that the blue "horse" bounding boxes in the rain image and the deraining image by GCANet[21] are also misclassified. These comparisons show that our deraining method is able to improve the accuracy of object detection in heavy rain scenarios in the real world against state-of-the-art methods.

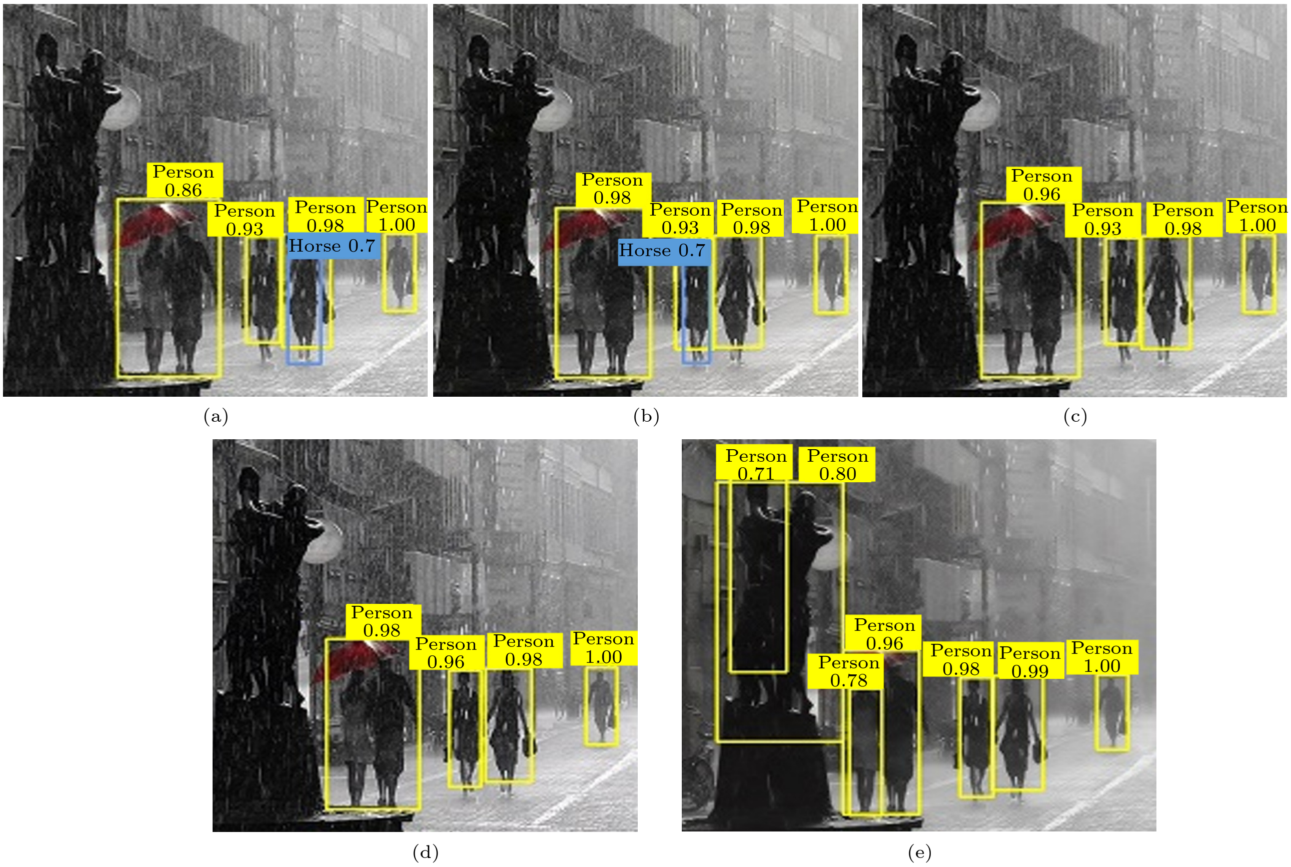


Fig.17. An application example of the proposed method for object detection in real rain weather. (a) Detection results of rainy images. (b) Detection results of deraining images of GCANet[21]. (c) Detection results of deraining images of NLEDN[16]. (d) Detection results of deraining images of PReNet[23]. (e) Detection results of deraining images of our method. More examples are displayed online②.

①https://drive.google.com/drive/folders/11cnn0qk6AWZmjc2QEd5OY9mD4dgZltrA, Mar. 2023.

②https://pan.baidu.com/s/1ki1adX6-CTqTqFkytS46WA?pwd=2115, Mar. 2023.

5.9 Model Parameters and Running Time

We further examine the efficiency of the proposed method. Table 8 shows that the parameters, used tool, and running time of the proposed method are comparable to those of the NLEDN[16] network. In addition, we make a comparison of average running time for images of size 512×512 with state-of-the-art methods. It can be observed that the average running time of the proposed method is less than half the time of JORDER[18]. All experiments are implemented on a PC with an Intel® Core$^{\mathrm{TM}}$ i7-7700K CPU@ 4.20 GHz, 64 RAM, and an NVIDIA GTX 1080Ti GPU.

Table 8. Comparisons of Model Size and Average Running Time

Method	GPU/CPU	Tool	Time (s)	Parameter (M)
DerainNet[12]	GPU	Matlab	0.28	0.75
JORDER[18]	GPU	Matlab	3.34	0.37
DDN[19]	GPU	Matlab	0.74	0.06
RGN[32]	GPU	Python	0.11	0.04
NLEDN[16]	GPU	Python	0.52	1.01
RESCAN[24]	GPU	Python	0.74	0.20
DID-MDN[25]	GPU	Python	0.20	0.37
GCANet[21]	GPU	Python	0.27	0.70
PReNet[23]	GPU	Python	0.26	0.17
Ours	GPU	Python	1.51	1.26

6 Conclusions

In this paper, a method using a residual channel attention network for single image deraining was proposed. By using the channel attention mechanism and the perceptual constraint in the feature space, the proposed network can not only capture the most important structures and details but also generate more realistic images. We also proposed an unsupervised finetuning approach to overcome the problem that existing deep learning based methods cannot generalize well on real rain images. Furthermore, our method can be flexibly extended to the tasks of handling heavy rain images with a haze-like effect and rainy videos. Extensive experimental results on synthetic and real-world images demonstrated that the proposed algorithm performs favorably against state-of-the-art methods. Compared with the second best method, our method improves 2.15 dB on the Rain100L dataset and 0.51 dB on the Rain100H dataset, respectively. In the future, we will consider extending the proposed method to adapt to downstream tasks, such as object detection, semantic segmentation, and person re-identification.

References

[1] Zhang D, Zhang H, Tang J, Wang M, Hua X, Sun Q. Feature pyramid transformer. In *Proc. the 16th European Conference on Computer Vision*, August 2020, pp.323–339. DOI: 10.1007/978-3-030-58604-1_20.

[2] Tang H, Li Z, Peng Z, Tang J. BlockMix: Meta regularization and self-calibrated inference for metric-based meta-learning. In *Proc. the 28th ACM International Conference on Multimedia*, October 2020, pp.610–618. DOI: 10.1145/3394171.3413884.

[3] Zhang D, Zhang H, Tang J, Hua X, Sun Q. Causal intervention for weakly-supervised semantic segmentation. In *Proc. Annual Conference on Neural Information Processing Systems*, December 2020, pp.655–666. DOI: 10.5555/3495724.3495780.

[4] Li Z, Sun Y, Zhang L, Tang J. CTNet: Context-based tandem tetwork for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(12): 9904–9917. DOI: 10.1109/TPAMI.2021.3132068.

[5] Kang L, Lin C, Fu Y. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 2012, 21(4): 1742–1755. DOI: 10.1109/TIP.2011.2179057.

[6] Luo Y, Xu Y, Ji H. Removing rain from a single image via discriminative sparse coding. In *Proc. the 15th International Conference on Computer Vision*, April 2015, pp.3397–3405. DOI: 10.1109/ICCV.2015.388.

[7] Zhu L, Fu C, Lischinski D, Heng P. Joint bi-layer optimization for single-image rain streak removal. In *Proc. the 16th International Conference on Computer Vision*, October 2017, pp.2545–2553. DOI: 10.1109/ICCV.2017.276.

[8] Wang Y, Liu S, Chen C, Xie D, Zeng B. Rain removal by image quasisparsity priors. arXiv: 1812.08348, 2018. https://arxiv.org/abs/1812.08348, Dec. 2018.

[9] Deng L, Huang T, Zhao X, Jiang T. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 2018, 59(7): 662–679. DOI: 10.1016/j.apm.2018.03.001.

[10] Du S, Liu Y, Ye M, Xu Z, Li J, Liu J. Single image deraining via decorrelating the rain streaks and background scene in gradient domain. *Pattern Recognition*, 2018, (79): 303–317. DOI: 10.1016/j.patcog.2018.02.016.

[11] Li Y, Tan T, Guo X, Lu J, Brown M. Rain streak removal using layer priors. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.2736–2744. DOI: 10.1109/CVPR.2016.299.

[12] Fu X, Huang J, Ding X, Liao Y, Paisley J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 2017, 26(6): 2944–2956. DOI: 10.1109/TIP.2017.2691802.

[13] Pan J, Liu S, Sun D, Zhang J, Liu Y, Ren J, Li Z, Tang J, Lu H, Tai Y, Yang M. Learning dual convolutional neural networks for low-level vision. In *Proc. the 31st IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.3070–3079. DOI: 10.1109/CVPR.2018.00324.

[14] Wang Y, Zhao X, Jiang T, Deng L, Chang Y, Huang T. Rain streak removal for single image via kernel guided CNN. arXiv: 1808.08545, 2018. https://arxiv.org/abs/1808.08545, Aug. 2018.

[15] Ye H, Li X, Liu H, Shi W, Liu M, Sun Q. Self-rening deep symmetry enhanced network for rain removal. In *Proc. the 25th IEEE International Conference on Image Processing*, September 2018, pp.2786–2790. DOI: 10.1109/ICIP.2019.8803265.

[16] Li G, He X, Zhang W, Chang H, Dong L, Lin L. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proc. the 26th ACM International Conference on Multimedia*, October 2018, 1056–1064. DOI: 10.1145/3240508.3240636.

[17] Li S, Ren W, Zhang J, Yu J, Guo X. Single image rain removal via a deep decomposition-composition network. *Computer Vision and Image Understanding*, 2019, 186: 48–57. DOI: 10.1016/j.cviu.2019.05.003.

[18] Yang W, Tan T, Feng J, Guo Z, Yan S, Liu J. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(6): 1377–1393. DOI: 10.1109/TPAMI.2019.2895793.

[19] Fu X, Huang J, Zeng D, Huang Y, Ding X, Paisley J. Removing rain from single images via a deep detail network. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, June 2017, pp.1715–1723. DOI: 10.1109/CVPR.2017.186.

[20] Fu X, Liang B, Huang Y, Ding X, Paisley J. Lightweight pyramid networks for image deraining. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(6): 1794–1807. DOI: 10.1109/TNNLS.2019.2926481.

[21] Chen D, He M, Fan Q, Liao J, Zhang L, Hou D, Yuan L, Hua G. Gated context aggregation network for image dehazing and deraining. In *Proc. the 2019 IEEE Winter Conference on Applications of Computer Vision*, January 2019, pp.1375–1383. DOI: 10.1109/WACV.2019.00151.

[22] Fu X, Qi Q, Huang Y, Ding X, Wu F, Paisley J. A deep tree-structured fusion model for single image deraining. arXiv: 1811.08632, 2018. https://arxiv.org/abs/1811.08632, Nov. 2018.

[23] Ren D, Zuo W, Hu Q, Zhu P, Meng D. Progressive image deraining networks: A better and simpler baseline. In *Proc. the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp.3937–3946. DOI: 10.1109/CVPR.2019.00406.

[24] Li X, Wu J, Lin Z, Liu H, Zha H. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proc. the 32nd European Conference on Computer Vision*, September 2018, pp.262–277. DOI: 10.1007/978-3-030-01234-2_16.

[25] Zhang H, Patel V. Density-aware single image de-raining using a multi-stream dense network. In *Proc. the 31st IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.695–704. DOI: 10.1109/CVPR.2018.00079.

[26] Zhang H, Sindagi V, Patel V. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020, 30(11): 3943–3956. DOI: 10.1109/TCSVT.2019.2920407.

[27] Qian R, Tan T, Yang W, Su J, Liu J. Attentive generative adversarial network for raindrop removal from a single image. In *Proc. the 31st Conference on Computer Vision and Pattern Recognition*, September 2018, pp.2482–2491. DOI: 10.1109/CVPR.2018.00263.

[28] Pu J, Chen X, Zhang L, Zhou Q, Zhao Y. Removing rain based on a cycle generative adversarial network. In *Proc. the 13th IEEE Conference on Industrial Electronics and Applications*, May 31–June 2, 2018, pp.621–626. DOI: 10.1109/ICIEA.2018.8397790.

[29] Pan J, Liu Y, Dong J, Zhang J, Ren J, Tang J, Tai Y, Yang M. Physics-based generative adversarial models for image restoration and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 43(7): 2449–2462. DOI: 10.1109/TPAMI.2020.2969348.

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.770–778. DOI: 10.1109/CVPR.2016.90.

[31] Wang D, Tang H, Pan J, Tang J. Learning a tree-structured channel-wise refinement network for efficient image deraining. In *Proc. IEEE International Conference on Multimedia and Expo*, July 2021. DOI: 10.1109/ICME51207.2021.9428187.

[32] Fan Z, Wu H, Fu X, Huang Y, Ding X. Residual guide network for single image deraining. In *Proc. the 26th ACM International Conference on Multimedia*, October 2018, pp.1751–1759. DOI: 10.1145/3240508.3240694.

[33] Wei W, Meng D, Zhao Q, Xu Z, Wu Y. Semi-supervised transfer learning for image rain removal. In *Proc. the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp.3877–3886. DOI: 10.1109/CVPR.2019.00400.

[34] Lin H, Li Y, Fu X, Ding X, Huang Y, Paisley J. Rain O'er Me: Synthesizing real rain to derain with data distillation. *IEEE Transactions on Image Processing*, 29(6): 7668–7680. DOI: 10.1109/TIP.2020.3005517.

[35] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In *Proc. the 31st IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.7132–7141. DOI: 10.1109/CVPR.2018.00745.

[36] Woo S, Park J, Lee J, Kweon I. CBAM: Convolutional block attention module. In *Proc. the 32nd European Conference on Computer Vision*, September 2018, pp.3–19. DOI: 10.1007/978-3-030-01234-21.

[37] Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, JohannesTotz, Wang Z,

Shi W. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, June 2017, pp.105–114. DOI: 10.1109/CVPR.2017.19.

[38] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A, Li F. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015, 115(3): 211–252. DOI: 10.1007/s11263-015-0816-y.

[39] Ledig C, Theis L, Caballero J, Cunningham A, Acosta A, Tejani A, Totz J, Wang Z, Shi W. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, June 2017, pp.105–114. DOI: 10.1109/CVPR.2017.19.

[40] Wang T, Yang X, Xu K, Chen S, Zhang Q, Lau R. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proc. the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, June 2019, pp.12270–12279. DOI: 10.1109/CVPR.2019.01255.

[41] Quan H, Mohammed G. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 2008, 44(13): 800–801. DOI: 10.1109/CVPR.2017.19.

[42] Wang Z, Simoncelli E, Bovik A. Multiscale structural similarity for image quality assessment. In *Proc. the 37th Asilomar Conference on Signals, Systems & Computers*, Nov. 2003, pp.1398–1402. DOI: 10.1109/ACSSC.2003.1292216.

[43] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv: 1412.6980, 2014. https://arxiv.org/abs/1412.6980, Jan. 2017.

[44] He K, Sun J, Tang X. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, 33(12): 2341–2353. DOI: 10.1109/TPAMI.2010.168.

[45] Li R, Cheong L, Tan T. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proc. the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, June 2020, pp.1633–1642. DOI: 10.1109/CVPR.2019.00173.

[46] Liu J, Yang W, Yang S, Guo Z. Erase or fill? Deep joint recurrent rain removal and reconstruction in videos. In *Proc. the 31st IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.3233–3242. DOI: 10.1109/CVPR.2018.00341.

[47] Tang H, Yuan C, Li Z, Tang J. Learning attention-guided pyramidal features for few-shot fine-grained recognition. *Pattern Recognition*, 2022. DOI: 10.1016/j.patcog.2022.108792. (preprint)

[48] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

[49] Lin T, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick C. Microsoft COCO: Common objects in context. In *Proc. the 13th European Conference on Computer Vision*, September 2014, pp.740–755. DOI: 10.1007/978-3-319-10602-148.

[50] Everingham M, Gool L, Williams C, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010, 88(2): 303–338. DOI: 10.1007/s11263-009-0275-4.

Di Wang is a Ph.D. candidate at the School of Software Technology, Dalian University of Technology, Dalian. She received her B.S. degree in digital media technology from Liaoning Normal University, Dalian, in 2018, and her M.S. degree in computer science and technology from Nanjing University of Science and Technology, Nanjing, in 2021. Her research interests include computer vision and deep learning.

Jin-Shan Pan is a professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. He received his Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, in 2017. His research interests include deblurring, image/video analysis and enhancement, and related vision problems. He is a member of IEEE.

Jin-Hui Tang is a professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. He received his B.E. degree in communication engineering and Ph.D. degree in signal and information processing from University of Science and Technology of China, Hefei, in 2003 and 2008, respectively. He received the Best Student Paper Award in MMM 2016, and Best Paper Awards in ACM MM 2007, PCM 2011, and ICIMCS 2011. He is a distinguished member of CCF, a senior member of IEEE and a member of ACM.